Internet Applications

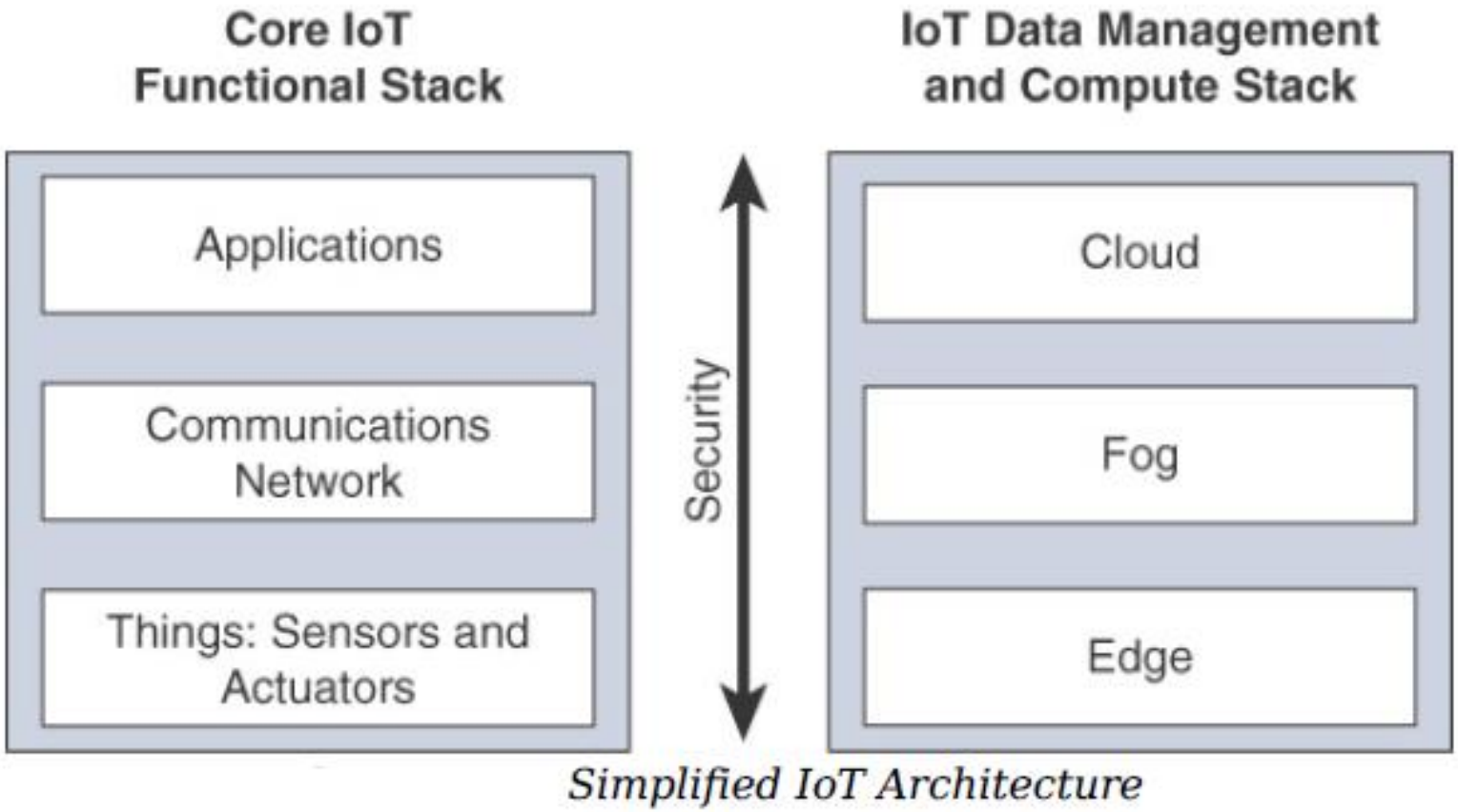
Dr./ Ahmed Mohamed Rabie

Chapter 1

Introduction

Internet ofThings

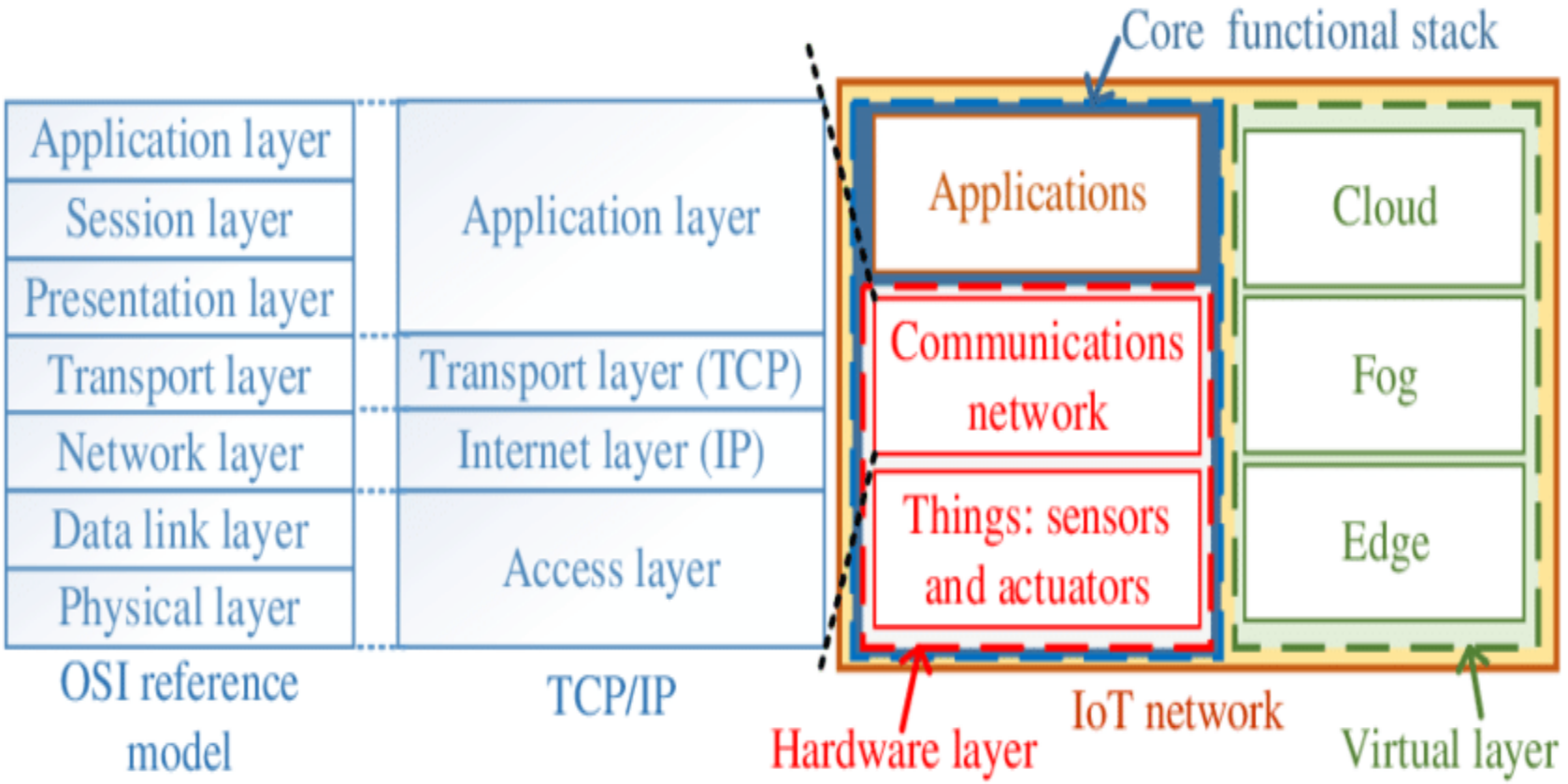
A Simplified IoT Architecture: Although considerable differences exist between the aforementioned reference models, they each approach IoT from a layered perspective, allowing development of technology and standards somewhat independently at each level or domain. The commonality between these frameworks is that they all recognize the interconnection of the IoT endpoint devices to a network that transports the data where it is ultimately used by applications, whether at the data center, in the cloud, or at various management points throughout the stack.



Simplified IoT Architecture

In fact, it can be noted that IoT architectures may differ somewhat depending on the industry use case or technology being deployed, and each has merit in solving the IoT heterogeneity problem discussed earlier. We present an IoT framework that highlights the fundamental building blocks that are common to most IoT systems and which is intended to help you in designing an IoT network.

This framework is presented as two parallel stacks: The **IoT Data Management and Compute Stack and the Core IoT Functional Stack**. Reducing the framework down to a pair of three-layer stacks in no way suggests that the model lacks the detail necessary to develop a sophisticated IoT strategy. Rather, the intention is to simplify the IoT architecture into its most basic building blocks and then to use it as a foundation to understand key design and deployment principles that are applied to industry-specific use cases. All the layers of more complex models are still covered, but they are grouped here in functional blocks that are easy to understand.


Application layer
Session layer
Presentation layer
Transport layer
Network layer
Data link layer
Physical layer
OSI reference model
Application layer
Transport layer (TCP)
Internet layer (IP)
Access layer
TCP/IP
Core functional stack
Applications
Communications network
Things: sensors and actuators
Cloud
Fog
Edge
Hardware layer
IoT network
Virtual layer

Nearly every published IoT model includes core layers similar to those shown on the left side , including “things,” a communications network, and applications. However, unlike other models, the framework presented here separates the core IoT and data management into parallel and aligned stacks, allowing you to carefully examine the functions of both the network and the applications at each stage of a complex IoT system. This separation gives you better visibility into the functions of each layer.

The presentation of the Core IoT Functional Stack in three layers is meant to simplify your understanding of the IoT architecture into its most foundational building blocks. Of course, such a simple architecture needs to be expanded on. The network communications layer of the IoT stack itself involves a significant amount of detail and incorporates a vast array of technologies.

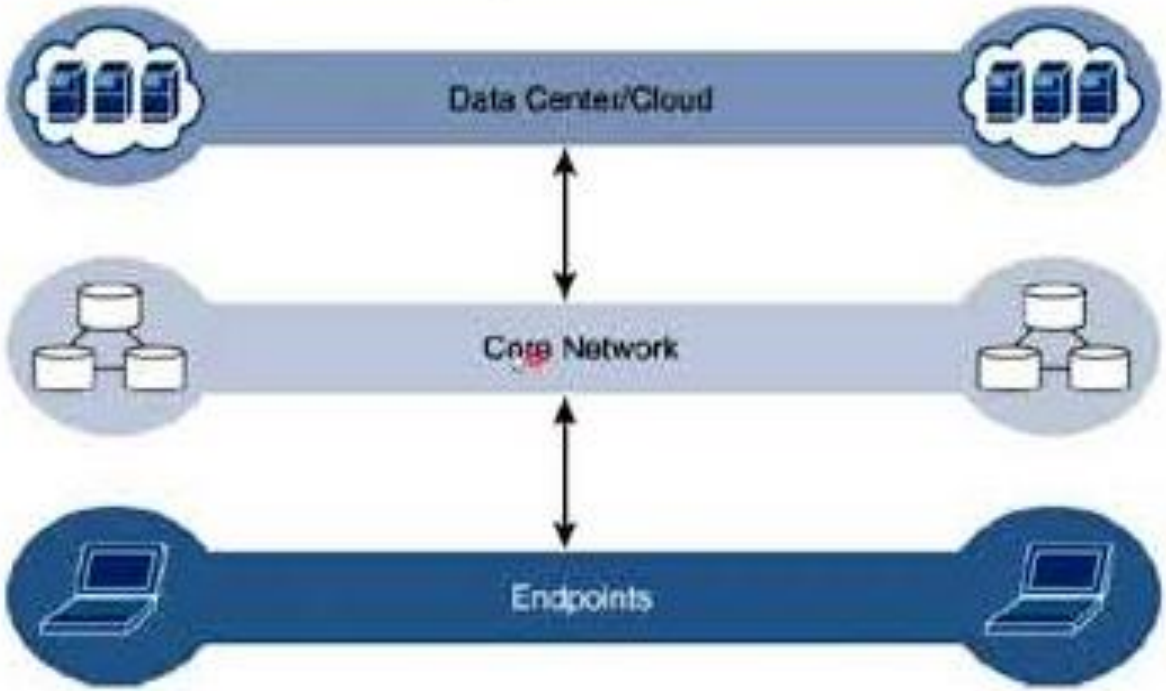
Consider for a moment the heterogeneity of IoT sensors and the many different ways that exist to connect them to a network. The network communications layer needs to consolidate these together, offer gateway and backhaul technologies, and ultimately bring the data back to a central location for analysis and processing.

Many of the last-mile technologies used in IoT are chosen to meet the specific requirements of the endpoints and are unlikely to ever be seen in the IT domain. However, the network between the gateway and the data center is composed mostly of traditional technologies that experienced IT professionals would quickly recognize. These include tunneling and VPN technologies, IP-based quality of service (QoS), conventional Layer 3 routing protocols such as BGP and IP-PIM, and security capabilities such as encryption, access control lists (ACLs), and firewalls.

Unlike with most IT networks, the applications and analytics layer of IoT doesn't necessarily exist only in the data center or in the cloud. Due to the unique challenges and requirements of IoT, it is often necessary to deploy applications and data management throughout the architecture in a tiered approach, allowing data collection, analytics, and intelligent controls at multiple points in the IoT system.

Data management is aligned with each of the three layers of the Core IoT Functional Stack. The three data management layers are the edge layer (data management within the sensors themselves), the fog layer (data management in the gateways and transit network), and the cloud layer (data management in the cloud or central data center).

IoT Data Management and Compute Stack



The Core IoT Functional Stack

IoT networks are built around the concept of "things," or smart objects performing functions and delivering new connected services. These objects are "smart" because they use a combination of contextual information and configured goals to perform actions. These actions can be self-contained (that is, the smart object does not rely on external systems for its actions); however, in most cases, the "thing" interacts with an external system to report information that the smart object collects, to exchange with other objects, or to interact with a management platform.

In this case, the management platform can be used to process data collected from the smart object and also guide the behavior of the smart object. From an architectural standpoint, several components have to work together for an IoT network to be operational:

"Things" layer: At this layer, the physical devices need to fit the constraints of the environment in which they are deployed while still being able to provide the information needed.

Communications network layer: When smart objects are not self-contained, they need to communicate with an external system. In many cases, this communication uses a wireless technology. This layer has four sublayers:

Access network sublayer: The last mile of the IoT network is the access network. This is typically made up of wireless technologies such as 802.11ah, 802.15.4g, and LoRa. The sensors connected to the access network may also be wired.

Gateways and backhaul network sublayer: A common communication system organizes multiple smart objects in a given area around a common gateway. The gateway communicates directly with the smart objects. The role of the gateway is to forward the collected information through a longer-range medium (called the backhaul) to a headend central station where the information is processed. This information exchange is a Layer 7 (application) function, which is he reason this object is called a gateway. On IP networks, this gateway also forwards packets from one IP network to another, and it therefore acts as a router.

Network transport sublayer: For communication to be successful, network and transport layer protocols such as IP and UDP must be implemented to support the variety of devices to connect and media to use.

IoT network management sublayer: Additional protocols must be in place to allow the headend applications to exchange data with the sensors. Examples include CoAP and MQTT.

Application and analytics layer: At the upper layer, an application needs to process the collected data, not only to control the smart objects when necessary, but to make intelligent decision based on the information collected and, in turn, instruct the "things" or other systems to adapt to the analyzed conditions and change their behaviors or parameters.

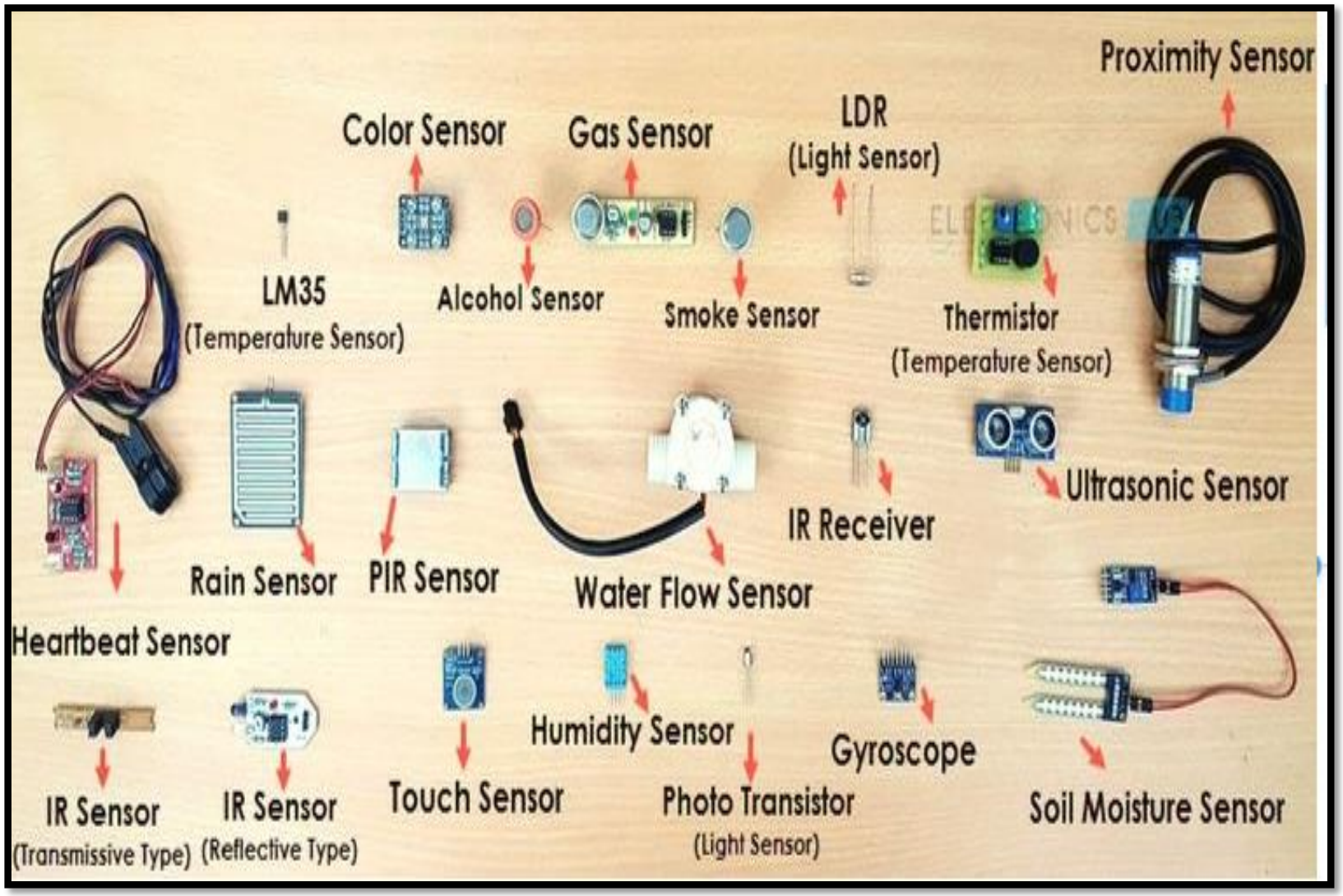
The following sections examine these elements and help you architect your IoT communication network.

Layer 1: Things: Sensors and Actuators Layer

Most IoT networks start from the object, or "thing," that needs to be connected. The 'Things' in IoT," provides more in-depth information about smart objects. From an architectural standpoint, the variety of smart object types, shapes, and needs drive the variety of IoT protocols and architectures. There are myriad ways to classify smart objects. One architectural classification could be:

Sensors and Actuators Layer

a) Battery-powered or power-connected: This classification is based on whether the object carries its own energy supply or receives continuous power from an external power source. Battery powered things can be moved more easily than line-powered objects. However, batteries limit the lifetime and amount of energy that the object is allowed to consume, thus driving transmission range and frequency.


Proximity Sensor
Color Sensor
Gas Sensor
LDR
(Light Sensor)
LM35
(Temperature Sensor)
Alcohol Sensor
Smoke Sensor
Thermistor
(Temperature Sensor)
Ultrasonic Sensor
IR Receiver
Rain Sensor
PIR Sensor
Water Flow Sensor
Heartbeat Sensor
Humidity Sensor
Gyroscope
IR Sensor
(Transmissive Type)
IR Sensor
(Reflective Type)
Touch Sensor
Photo Transistor
(Light Sensor)
Soil Moisture Sensor

b) Mobile or static: This classification is based on whether the "thing" should move or always stay at the same location. A sensor may be mobile because it is moved from one object to another (for example, a viscosity sensor moved from batch to batch in a chemical plant) or because it is attached to a moving object (for example, a location sensor on moving goods in a warehouse or factory floor). The frequency of the movement may also vary, from occasional to permanent. The range of mobility (from a few inches to miles away) often drives the possible power source.

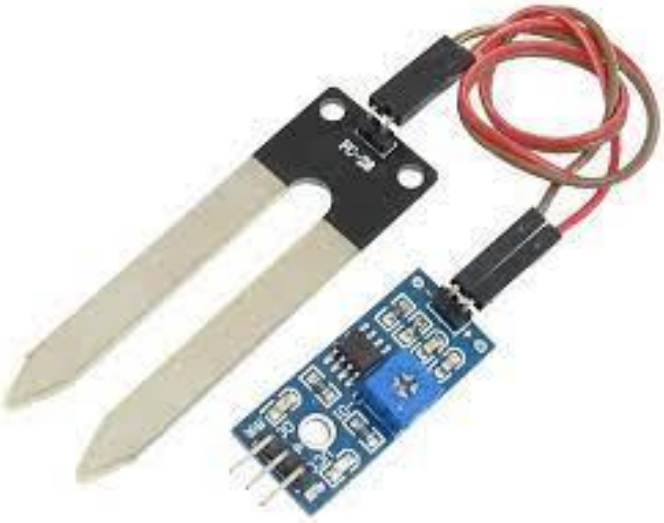
c) Low or high reporting frequency: This classification is based on how often the object should report monitored parameters. A rust sensor may report values once a month. A motion sensor may report acceleration several hundred times per second. Higher frequencies drive higher energy consumption, which may create constraints on the possible power source (and therefore the object mobility) and the transmission range.

Rust Sensor

Motion Sensor

d) Simple or rich data: This classification is based on the quantity of data exchanged at each report cycle. A humidity sensor in a field may report a simple daily index value (on a binary scale from 0 to 255), while an engine sensor may report hundreds of parameters, from temperature to pressure, gas velocity, compression speed, carbon index, and many others. Richer data typically drives higher power consumption.

Humidity Sensor

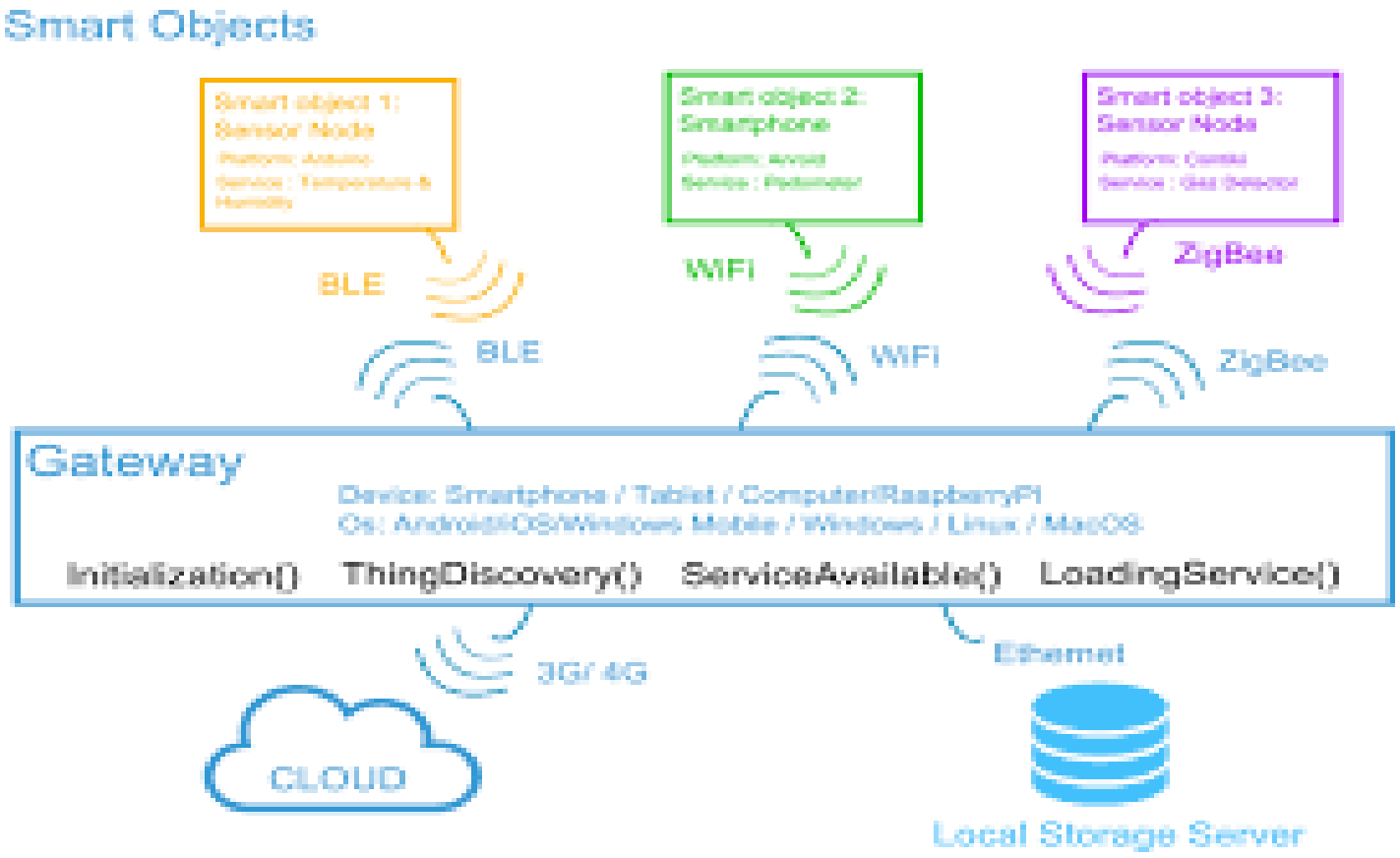


Engine Sensor

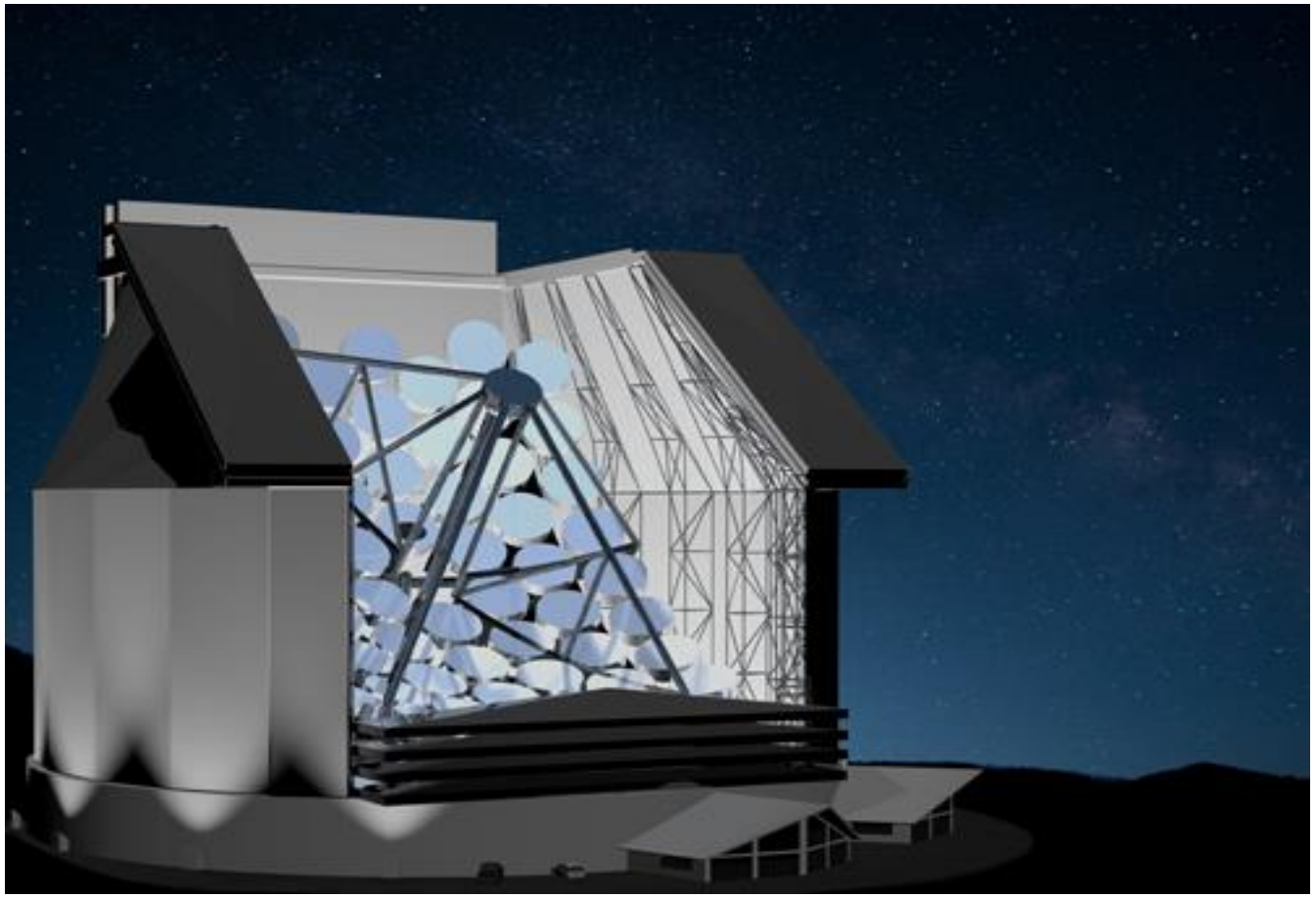
This classification is often combined with the previous to determine the object data throughput (low throughput to high throughput). You may want to keep in mind that throughput is a combined metric. A medium-throughput object may send simple data at rather high frequency (in which case the flow structure looks continuous), or may send rich data at rather low frequency (in which case the flow structure looks bursty).

e) Report range: This classification is based on the distance at which the gateway is located. For example, for your fitness band to communicate with your phone, it needs to be located a few meters away at most. The assumption is that your phone needs to be at visual distance for you to consult the reported data on the phone screen. If the phone is far away, you typically do not use it, and reporting data from the band to the phone is not necessary. By contrast, a moisture sensor in the asphalt of a road may need to communicate with its reader several hundred meters or even kilometers away.

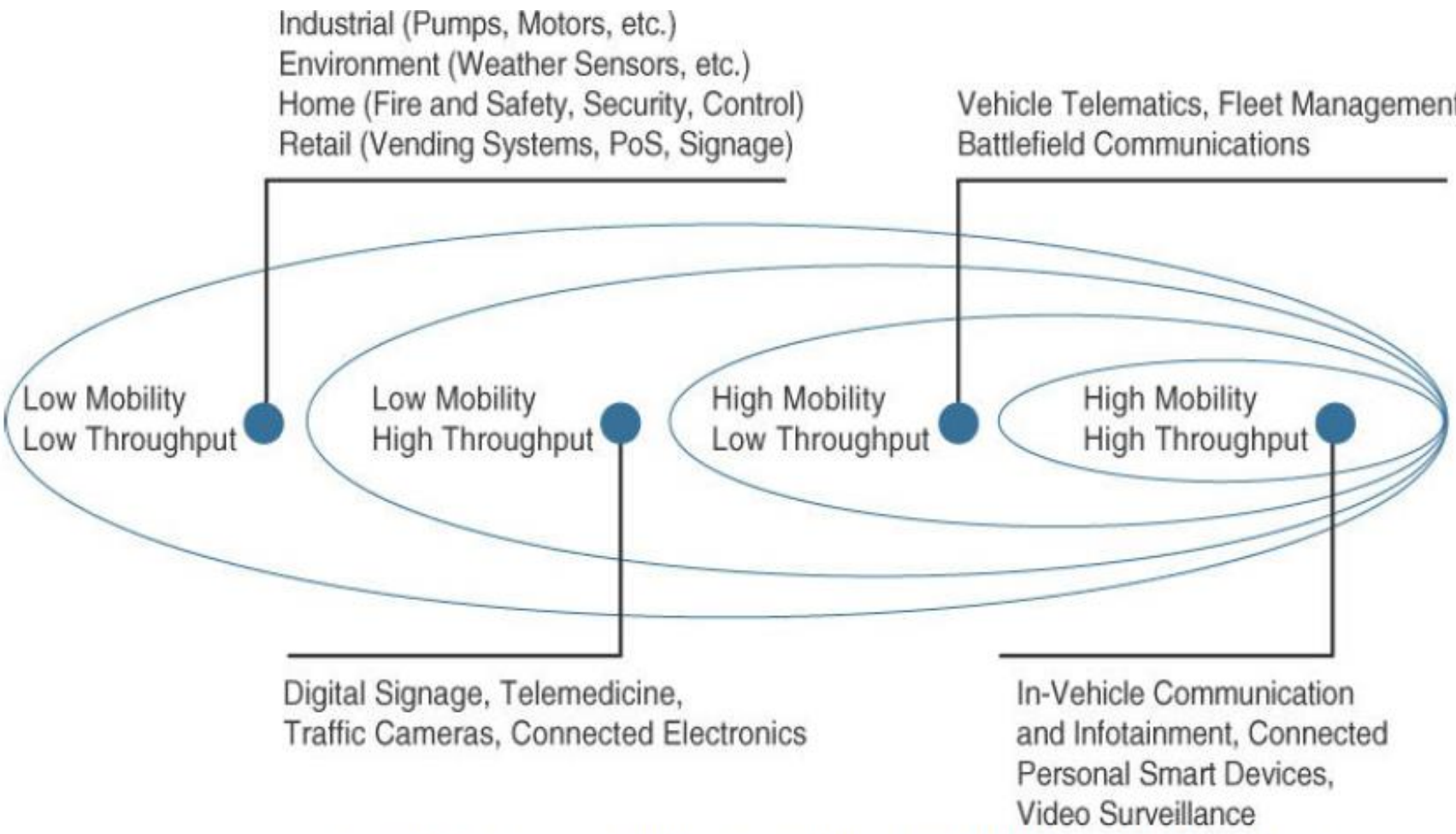
f) Object density per cell: This classification is based on the number of smart objects (with a similar need to communicate) over a given area, connected to the same gateway. An oil pipeline may utilize a single sensor at key locations every few miles. By contrast, telescopes like the SETI Colossus telescope at the Whipple Observatory deploy hundreds, and sometimes thousands, of mirrors over a small area, each with multiple gyroscopes, gravity, and vibration sensors.


Smart Objects
Smart object 1:
Sensor Node
Smart object 2:
Smartphone
Smart object 3:
Sensor Node
BLE
WiFi
ZigBee
BLE
WiFi
ZigBee
Gateway
Device: Smartphone / Tablet / Computer/RaspberryPI
Os: Android/iOS/Windows Mobile / Windows / Linux / MacOS
Initialization()
ThingDiscovery()
ServiceAvailable()
LoadingService()
3G/ 4G
Ethernet
CLOUD
Local Storage Server

SETI Colossus telescope



From a network architectural standpoint, your initial task is to determine which technology should be used to allow smart objects to communicate. This determination depends on the way the "things" are classified. However, some industries (such as manufacturing and utilities) may include objects in various categories, matching different needs. It provides some examples of applications matching the combination of mobility and throughput requirements.



Example of Sensor Applications Based on Mobility and Throughput

The categories used to classify things can influence other parameters and can also influence one another. For example, a battery-operated highly mobile object (like a heart rate monitor, for example) likely has a small form factor. A small sensor is easier to move or integrate into its environment. At the same time, a small and highly mobile smart object is unlikely to require a large antenna and a powerful power source.

This constraint will limit the transmission range and, therefore, the type of network protocol available for its connections. The criticality of data may also influence the form factor and, therefore, the architecture. For example, a missing monthly report from an asphalt moisture sensor may simply flag an indicator for sensor (or battery) replacement. A multi-mirror gyroscope report missing for more than 100 ms may render the entire system unstable or unusable.

These sensors either need to have a constant source of power (resulting in limited mobility) or need to be easily accessible for battery replacement (resulting in limited transmission range). A first step in designing an IoT network is to examine the requirements in terms of mobility and data transmission (how much data, how often).